PRE-APPEAL BRIEF REQUEST FOR REVIEW

Docket Number
20423-08166

Pursuant to 240 OG 45 and the *Legal Framework For EFS-Web*, I hereby certify that this follow-on correspondence is being officially submitted through the USPTO EFS-Web system from the Pacific Time Zone of the United States on the local date shown below.

on February 20, 2008

Signature /Nikhil Iyengar/

Typed or printed name

Nikhil Iyengar

Application Number	Filed
10/763,673	January 22, 2004

First Named Inventor: Frederic Perriot

Art Unit	Examiner
2135	Randal D. Moran

This request is being filed with a notice of appeal.

I am the

☐ applicant/inventor.

☐ assignee of record of the entire interest.
See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed.

☒ attorney or agent of record.
Registration number 60,910.

☐ attorney or agent acting under 37 CFR 1.34.
Registration number if acting under 37 CFR 1.34 _____

/Nikhil Iyengar/
Signature

Nikhil Iyengar
Typed or printed name

(415) 875-2367
Telephone number

February 20, 2008
Date

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below*.

☒ *Total of 1 of 1 forms is submitted.

ATTACHMENT TO THE

PRE-APPEAL BRIEF REQUEST FOR REVIEW

Pre-appeal review is requested because the rejections of record are clearly improper and without any factual or legal basis. Applicants respectfully request that the Panel indicate claims 1-19 and 24-27 recite allowable subject matter.

I. Status of the Claims

Claims 1-19 and 24-27 are pending and stand finally rejected. No claims have been amended since the last Office Action.

II. Rejection of claim 6 under 35 USC § 103(a)

Independent claim 6 is rejected under 35 USC § 103(a) as allegedly being unpatentable over U.S. Patent No. 5,881,151 to Yamamoto in view of U.S. Patent No. 5,826,013 to Nachenberg. Claim 6 recites identifying computer code **suspected of currently containing malicious code, the computer code having a decryption loop and a body**. The **decryption loop and body are optimized** to produce optimized code. The optimized code is subjected to a malicious code detection protocol, and a confirmation of malicious code is declared responsive to detecting malicious code. The claimed invention beneficially optimizes the code to simplify it so that malicious code detection protocols can be more efficiently and effectively applied to the code. For example, a virus may contain intentionally complexified code that makes virus detection protocols slow or inaccurate. The claimed invention first optimizes the code so that the detection protocols can be more successful.

Yamamoto discloses compiling and optimizing a virus-free source program and adding virus diagnosing (object) code to the resulting program (object) code (Yamamoto, col. 4, lines 21-46). The virus diagnosing code can be run later to verify that the program code has not been

modified by a virus subsequent to being compiled (Yamamoto, col. 6, lines 38-50). The diagnosing code operates under the assumption that the program code is free of viruses since it was compiled and optimized from the known source program. Yamamoto is not concerned with optimizing code that is currently suspected of currently containing malicious code. Nachenberg describes a method for detecting a polymorphic virus using emulation, similar to the method described in the Background Art of the present specification. Nachenberg does not disclose optimizing code that is suspected of currently containing malicious code.

Examiner does not address "identifying computer code suspected of currently containing malicious code, the computer code having a decryption loop and a body," as recited in the claim. This is an essential element of the claim needed for a prima facie rejection under MPEP 2142. Neither Yamamoto nor Nachenberg discloses this limitation.

Examiner cites col. 6, line 54 to col. 7, line 8 and the polymorphic anti-virus module (PAM) 200 of Nachenberg as disclosing "optimizing the decryption loop to produce optimized loop code" and "optimizing the body to produce optimized body code" However, this portion of Nachenberg merely discloses emulating, not optimizing, a portion of code to determine if the code contains a virus. Such emulation may reveal a virus decryption loop.

In an Advisory Action dated February 13, 2008, Examiner states that that static exclusion module 230 and the dynamic exclusion module 240 of Nachenberg disclose optimizing the decryption loop and optimizing the body code[1]. However, though these modules may "substantially reduce the number of file instructions that must be emulated," (Nachenberg, col. 6, lines 56-57) the modules do not optimize a decryption loop and body code. Rather, the static

---

[1] Applicant previously discussed this portion of Nachenberg. See Amendment B, filed January 22, 2008, page 10, lines 8-12.

exclusion module 230 merely looks at "gross features of the executable image 100 that are inconsistent with various polymorphic viruses." (Nachenberg, col. 7, lines 9-20). The static exclusion module 230 is concerned with analyzing file types, file sizes, and detecting the presence of certain instructions (Nachenberg, col. 7, lines 24-40), rather than optimizing a decryption loop and body code. The dynamic exclusion module 240 accesses instruction/interrupt usage profiles 224 of known viruses during emulation to determine whether the emulated code may be part of a virus decryption loop (Nachenberg, col. 6, line 65 – col. 7, line 2). Again, the dynamic exclusion module decreases number of instructions to be emulated by comparing the instructions to known virus profiles, not by optimizing a decryption loop and body code.

III. Rejection of claim 27 under 35 USC § 103(a)

Independent claim 27 is rejected under 35 USC § 103(a) as allegedly being unpatentable over Yamamoto in view of U.S. Patent Publication No. 2004/0221279 to Lovett. Claim 27 recites performing a dead code elimination procedure on the computer code, noting the **amount of dead code eliminated** during the procedure, and declaring a **suspicion of malicious code in the computer code when this amount exceeds a threshold**. The claimed invention enables the detection of suspicious code by determining that the code contains a certain amount of dead code, which is often found in malicious code.

Claim 27 is not disclosed by the combination of Yamamoto and Lovett. As discussed above, Yamamoto discloses optimizing only clean source code, not optimizing computer code to determine if there was malicious code in the computer code prior to the optimization (dead code elimination is a type of optimization). Lovett discloses dead code elimination but is not

concerned with declaring a suspicion of malicious code based on the amount of dead code eliminated.

Accordingly, the references do not disclose "when the amount of dead code eliminated during the dead code elimination procedure exceeds a preselected dead code threshold, declaring a suspicion of malicious code in the computer code." Paragraphs [0133], [0144], [0091], and [0098] cited by Examiner merely mention thresholds used in other contexts, such as an execution count threshold for group block generation or a profiling metric threshold for group block construction. These are not thresholds of amounts of dead code, and the thresholds are not used to declare a suspicion of malicious code. In paragraph [0091] of Lovett, dead code elimination is **triggered by** a profiling metric exceeding a certain threshold, teaching away from the claimed invention.

In the "Response to Arguments" section of the Office Action, Examiner further points to Lovett [0107] and Yamamoto, col. 6, lines 31-37, as disclosing the above element. However, Lovett [0107] merely discloses recording global dead code transformations in subject registers in order to prune IR (Intermediate Representation) trees. Yamamoto, col. 6, lines 31-37, merely discloses interrupting execution and creating an output after a virus is detected. Examiner also states that "dead code elimination is a profiling metric." However, Examiner does not provide support for this statement. Lovett provides examples of profiling metrics in paragraph [0093] and none of the examples include dead code elimination.

In an Advisory Action dated February 13, 2008, Examiner stated that a "global dead code transformations list" indicates the amount of dead code eliminated during a dead code elimination procedure. However, Lovett merely discloses that "the dead code transformations can be recorded as a list of 'dead' subject registers." (Lovett, para. [0107]). Even if a

"transformation" can be interpreted as an "elimination", the size of each transformation is not correlated to the recorded registers (for example, a very large transformation may be recorded as a single "dead" register), and Lovett provides no connection between the amount of dead code eliminated and the recorded list of subject registers. Additionally, neither Lovett nor Yamamoto discloses comparing an amount of dead code eliminated to a threshold.

IV. Summary

Based on the foregoing, Applicants respectfully submit that each of the pending rejections suffers from a clear deficiency in the prima facie case asserted in support of the rejection. Accordingly, Applicants request that the rejections of claims 1-19 and 24-27 be withdrawn.

Respectfully submitted,
Frederic Perriot

Dated: February 20, 2008

By: /Nikhil Iyengar/

Nikhil Iyengar, Attorney of Record
Registration No. 60,910
FENWICK & WEST LLP
555 California Street
San Francisco, CA 94104
Phone: (415) 875-2367
Fax: (415) 281-1350